

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1 1-66 (Canceled).

1 67. (Currently amended) ~~An~~ A computer-implemented authenticated-
2 encryption method that uses an n-bit block cipher, a key, and an n-bit nonce to
3 encrypt a message into a ciphertext, the method comprising:
4 partitioning the message into m-1 message blocks and one final fragment,
5 each message block having n bits and the final fragment having between 0 and n
6 bits;
7 using the block cipher, the key, and the nonce to generate a sequence of m
8 offsets, each offset having n bits, wherein the sequence of offsets is computed by
9 (a) computing a 0th basis offset by applying the block cipher, keyed by the key, to
10 a constant; (b) for each positive number i, defining the ith basis offset from the
11 prior basis offset by shifting the prior basis offset left one position, and then
12 xoring the resulting value with a constant that depends on the first bit of the prior
13 basis offset; (d) computing a base offset by applying the block cipher, keyed by
14 the key, to the xor of the 0th basis offset and the nonce; (e) defining the 1st offset
15 in the sequence of offsets as the xor of the 0th basis offset and the base offset; and
16 (f) for each integer i between two and m, defining the ith offset in the sequence of
17 offsets as the xor of the prior offset and the jth basis offset, where j is the number
18 of zero-bits following the last one-bit when the number i is written in binary;

19 using the block cipher, the key, the nonce, and the length of the message to
20 generate an n-bit final offset;
21 for each number i between 1 and m-1, xoring the i^{th} message block with
22 the i^{th} offset to determine an i^{th} input block;
23 for each number i between 1 and m-1, applying the block cipher, keyed by
24 the key, to the i^{th} input block, to determine an i^{th} output block;
25 for each number i between 1 and m-1, xoring the i^{th} output block with the
26 i^{th} offset to determine an i^{th} ciphertext block;
27 concatenating the m-1 ciphertext blocks to determine a ciphertext body;
28 computing an encoded length by encoding the length of the final fragment
29 as an n-bit string;
30 xoring the encoded length with the final offset to determine a precursor
31 pad;
32 computing a pad by applying the block cipher, keyed by the key, to the
33 precursor pad;
34 xoring the final fragment with a portion of the pad to determine a
35 ciphertext fragment having the same length as the final fragment;
36 computing a padded ciphertext fragment by appending to the ciphertext
37 fragment a sufficient number of zero bits so that the padded ciphertext fragment
38 has n bits;
39 computing a checksum by xoring together the m-1 message blocks, the
40 pad, and the padded ciphertext fragment;
41 computing a precursor full tag by xoring together the checksum and the
42 m^{th} offset;
43 determining a full tag by applying the block cipher, keyed by the key, to
44 the precursor full tag;
45 computing a tag as a portion of the full tag; and

46 defining the ciphertext to be the ciphertext body, the ciphertext fragment,
47 and the tag.

1 68. (Previously presented) A computer-readable storage medium storing
2 instructions that when executed by a computer cause the computer to perform an
3 authenticated-encryption method that uses an n -bit block cipher, a key, and an n -
4 bit nonce to encrypt a message into a ciphertext, the method comprising:

5 partitioning the message into $m-1$ message blocks and one final fragment,
6 each message block having n bits and the final fragment having between 0 and n
7 bits;

8 using the block cipher, the key, and the nonce to generate a sequence of m
9 offsets, each offset having n bits, wherein the sequence of offsets is computed by
10 (a) computing a 0th basis offset by applying the block cipher, keyed by the key, to
11 a constant; (b) for each positive number i , defining the i^{th} basis offset from the
12 prior basis offset by shifting the prior basis offset left one position, and then
13 xoring the resulting value with a constant that depends on the first bit of the prior
14 basis offset; (d) computing a base offset by applying the block cipher, keyed by
15 the key, to the xor of the 0th basis offset and the nonce; (e) defining the 1st offset
16 in the sequence of offsets as the xor of the 0th basis offset and the base offset; and
17 (f) for each integer i between two and m , defining the i^{th} offset in the sequence of
18 offsets as the xor of the prior offset and the j^{th} basis offset, where j is the number
19 of zero-bits following the last one-bit when the number i is written in binary;

20 using the block cipher, the key, the nonce, and the length of the message to
21 generate an n -bit final offset;

22 for each number i between 1 and $m-1$, xoring the i^{th} message block with
23 the i^{th} offset to determine an i^{th} input block;

24 for each number i between 1 and $m-1$, applying the block cipher, keyed by
25 the key, to the i^{th} input block, to determine an i^{th} output block;

26 for each number i between 1 and $m-1$, xoring the i^{th} output block with the
 27 i^{th} offset to determine an i^{th} ciphertext block;
 28 concatenating the $m-1$ ciphertext blocks to determine a ciphertext body;
 29 computing an encoded length by encoding the length of the final fragment
 30 as an n -bit string;
 31 xoring the encoded length with the final offset to determine a precursor
 32 pad;
 33 computing a pad by applying the block cipher, keyed by the key, to the
 34 precursor pad;
 35 xoring the final fragment with a portion of the pad to determine a
 36 ciphertext fragment having the same length as the final fragment;
 37 computing a padded ciphertext fragment by appending to the ciphertext
 38 fragment a sufficient number of zero bits so that the padded ciphertext fragment
 39 has n bits;
 40 computing a checksum by xoring together the $m-1$ message blocks, the
 41 pad, and the padded ciphertext fragment;
 42 computing a precursor full tag by xoring together the checksum and the
 43 m^{th} offset;
 44 determining a full tag by applying the block cipher, keyed by the key, to
 45 the precursor full tag;
 46 computing a tag as a portion of the full tag; and
 47 defining the ciphertext to be the ciphertext body, the ciphertext fragment,
 48 and the tag.

1 | 69. (Currently amended) ~~An~~ A computer-implemented authenticated-
 2 encryption method that uses an n -bit block cipher, a key, and an n -bit nonce to
 3 | encrypt a message of arbitrary length into a ciphertext of the same length, the
 4 method comprising:

5 partitioning the message into $m-1$ message blocks and one final fragment,
6 each message block having n bits and the final fragment having between 0 and n
7 bits;
8 generating $m+1$ offsets using a sequence shift and xor operations, this
9 sequence of shift and xor operations being applied to a starting value determined
10 using the block cipher, the key, and the nonce;
11 for each number i between 1 and $m-1$, xoring the i^{th} message block with
12 the i^{th} offset to determine an i^{th} input block;
13 for each number i between 1 and $m-1$, applying the block cipher, keyed by
14 the key, to the i^{th} input block, to determine an i^{th} output block;
15 for each number i between 1 and $m-1$, xoring the i^{th} output block with the
16 i^{th} offset to determine an i^{th} ciphertext block;
17 concatenating the $m-1$ ciphertext blocks to determine a ciphertext body;
18 computing an encoded length by encoding the length of the final fragment
19 as an n -bit string;
20 xoring the encoded length with the m^{th} offset to determine a precursor pad;
21 computing a pad by applying the block cipher, keyed by the key, to the
22 precursor pad;
23 xoring the final fragment with a portion of the pad to determine a
24 ciphertext fragment having the same length as the final fragment;
25 computing a padded ciphertext fragment by appending to the ciphertext
26 fragment a sufficient number of zero bits so that the padded ciphertext fragment
27 has n bits;
28 computing a checksum by xoring together the $m-1$ message blocks, the
29 pad, and the padded ciphertext fragment;
30 computing a precursor full tag by xoring together the checksum and the
31 $(m+1)^{\text{st}}$ offset;

32 determining a full tag by applying the block cipher, keyed by the key, to
33 the precursor full tag;
34 computing a tag as a portion of the full tag; and
35 defining the ciphertext to be the ciphertext body, the ciphertext fragment,
36 and the tag.

1 70 (Currently amended) A computer-readable storage medium storing
2 instructions that when executed by a computer cause the computer to perform an
3 authenticated-encryption method that uses an n-bit block cipher, a key, and an n-
4 bit nonce to encrypt a message of an arbitrary length into a ciphertext of the same
5 length, the method comprising:
6 partitioning the message into m-1 message blocks and one final fragment,
7 each message block having n bits and the final fragment having between 0 and n
8 bits;
9 generating m+1 offsets using a sequence shift and xor operations, this
10 sequence of shift and xor operations being applied to a starting value determined
11 using the block cipher, the key, and the nonce;
12 for each number i between 1 and m-1, xoring the i^{th} message block with
13 the i^{th} offset to determine an i^{th} input block;
14 for each number i between 1 and m-1, applying the block cipher, keyed by
15 the key, to the i^{th} input block, to determine an i^{th} output block;
16 for each number i between 1 and m-1, xoring the i^{th} output block with the
17 i^{th} offset to determine an i^{th} ciphertext block;
18 concatenating the m-1 ciphertext blocks to determine a ciphertext body;
19 computing an encoded length by encoding the length of the final fragment
20 as an n-bit string;
21 xoring the encoded length with the m^{th} offset to determine a precursor pad;

22 computing a pad by applying the block cipher, keyed by the key, to the
23 precursor pad;
24 xoring the final fragment with a portion of the pad to determine a
25 ciphertext fragment having the same length as the final fragment;
26 computing a padded ciphertext fragment by appending to the ciphertext
27 fragment a sufficient number of zero bits so that the padded ciphertext fragment
28 has n bits;
29 computing a checksum by xoring together the $m-1$ message blocks, the
30 pad, and the padded ciphertext fragment;
31 computing a precursor full tag by xoring together the checksum and the
32 $(m+1)^{\text{st}}$ offset;
33 determining a full tag by applying the block cipher, keyed by the key, to
34 the precursor full tag;
35 computing a tag as a portion of the full tag; and
36 defining the ciphertext to be the ciphertext body, the ciphertext fragment,
37 and the tag.